

第2章Redis的安装与配置

0克隆主机

克隆母机完成后，启动克隆的主机，然后修改两个配置文件：

- 修改主机名：/etc/hostname
- 修改网络配置：/etc/sysconfig/network-scripts/ifcfg-ens33

1 Redis的安装

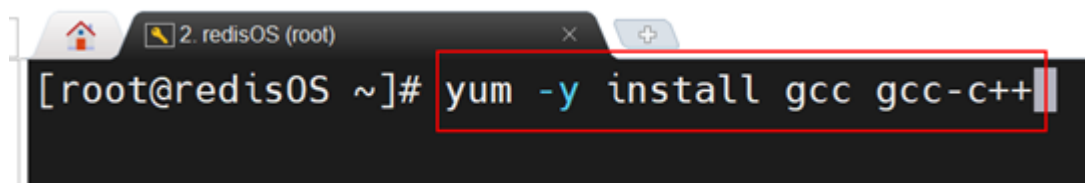
安装Redis

安装gcc

由于Redis是由C/C++语言编写的，而从官网下载的Redis安装包是需要编译后才可安装的，所以对其进行编译就必须使用相关编译器。对于C/C++语言的编译器，使用最多的是gcc与gcc-c++，而这两款编译器在CentOS7中是没有安装的，所以首先要安装这两款编译器。

gcc, GNU Compiler Collection, GNU 编译器集合

GNU, 即GNU计划



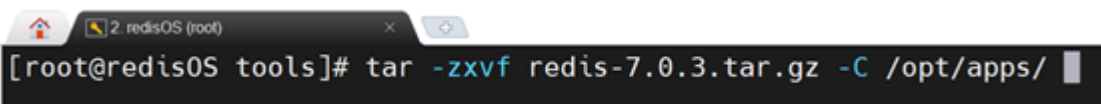
```
[root@redisOS ~]# yum -y install gcc gcc-c++
```

下载上传Redis

从官网 (<http://redis.io>) 下载Redis，并上传到Linux系统相应目录。

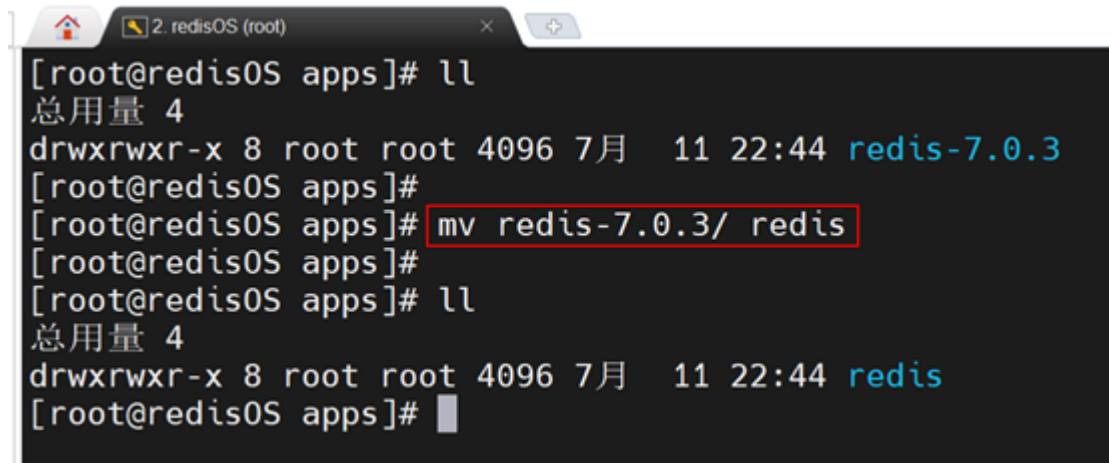
解压Redis

将Redis解压到需要的目录中。



```
[root@redisOS tools]# tar -zxvf redis-7.0.3.tar.gz -C /opt/apps/
```

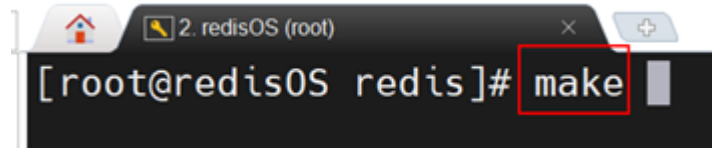
进入到/opt/apps目录中再将Redis解压包目录更名为redis。

A terminal window titled '2. redisOS (root)' showing a series of commands and their outputs. The user enters 'll' to list files, showing 'redis-7.0.3'. Then they enter 'mv redis-7.0.3/ redis', which is highlighted with a red box. Finally, they enter 'll' again, showing 'redis'.

```
[root@redis0S apps]# ll
总用量 4
drwxrwxr-x 8 root root 4096 7月  11 22:44 redis-7.0.3
[root@redis0S apps]#
[root@redis0S apps]# mv redis-7.0.3/ redis
[root@redis0S apps]#
[root@redis0S apps]# ll
总用量 4
drwxrwxr-x 8 root root 4096 7月  11 22:44 redis
[root@redis0S apps]#
```

编译

编译过程是根据Makefile文件进行的，而Redis解压包中已经存在该文件了，所以可以直接进行编译了。进入到解压目录中，然后执行编译命令make。

A terminal window titled '2. redisOS (root)' showing the user entering the 'make' command, which is highlighted with a red box.

```
[root@redis0S redis]# make
```

安装

在Linux中对于编译过的安装包执行make install进行安装。

```
1 2 redisOS (root)
[root@redisOS redis]# make install
cd src && make install
which: no python3 in (/usr/java/jdk1.8.0_161//bin:/u
bin:/usr/bin:/root/bin)
make[1]: 进入目录"/opt/apps/redis/src"
CC Makefile.dep
make[1]: 离开目录"/opt/apps/redis/src"
which: no python3 in (/usr/java/jdk1.8.0_161//bin:/u
bin:/usr/bin:/root/bin)
make[1]: 进入目录"/opt/apps/redis/src"

Hint: It's a good idea to run 'make test' ;)

INSTALL redis-server
INSTALL redis-benchmark
INSTALL redis-cli
make[1]: 离开目录"/opt/apps/redis/src"
[root@redisOS redis]#
```

Redis启动与停止

前台启动

在任意目录执行redis-server命令即可启动Redis。这种启动方式会占用当前命令行窗口。

```
1 2 redisOS (root)
[root@redisOS redis]# redis-server
6879:C 15 Jul 2022 12:32:01.386 # o000o000o000o Redis is starting o000o000o000o
6879:C 15 Jul 2022 12:32:01.386 # Redis version=7.0.3, bits=64, commit=00000000,
ed=0, pid=6879, just started
6879:C 15 Jul 2022 12:32:01.386 # Warning: no config file specified, using the d
config. In order to specify a config file use redis-server /path/to/redis.conf
6879:M 15 Jul 2022 12:32:01.386 * Increased maximum number of open files to 1003
as originally set to 1024).
6879:M 15 Jul 2022 12:32:01.386 * monotonic clock: POSIX clock_gettime

Redis 7.0.3 (00000000/0) 64 bit
Running in standalone mode
Port: 6379
PID: 6879

https://redis.io
```

再开启一个会话窗口，可以查看到当前的Redis进程，默认端口号为6379。

```
[root@redisOS ~]# ps aux | grep redis
root      12015  0.1  0.5 162512 10256 pts/0    Sl+  10:13   0:00 redis-server *:6379
root      12108  0.0  0.0 112824   988 pts/1    S+   10:14   0:00 grep --color=auto r
edis
[root@redisOS ~]#
```

通过Ctrl + C命令可以停止Redis。

命令式后台启动

使用nohub命令，最后再添加一个&符，可以使要启动的程序在后台以守护进程方式运行。这样的好处是，进程启动后不会占用一个会话窗口，且其还会在当前目录，即运行启动命令的当前目录中创建一个nohub.out文件用于记录Redis的操作日志。

```
[root@redisOS redis]# nohub redis-server &
[1] 6996
[root@redisOS redis]# nohub: 忽略输入并把输出追加到"nohub.out"
回车
[root@redisOS redis]#
```

Redis的停止

通过redis-cli shutdown命令可以停止Redis。

```
[root@redisOS redis]# redis-cli shutdown
[1]+ 完成 nohub redis-server
[root@redisOS redis]#
```

配置式后台启动

使用nohub命令可以使Redis后台启动，但每次都要键入nohub与&符，比较麻烦。可以通过修改Linux中Redis的核心配置文件redis.conf达到后台启动的目的。redis.conf文件在Redis的安装目录根下。将daemonize属性值由no改为yes，使Redis进程以守护进程方式运行。

```
303
304 ##### GENERAL #####
305
306 # By default Redis does not run as a daemon.
307 # Note that Redis will write a pid file in /v
308 # When Redis is supervised by upstart or syst
309 daemonize yes
310
```

修改后再启动Redis，就无需再键入nohup与&符号了，但必须要指定启动所使用的Redis配置文件。这是为什么呢？

使用nohup redis-server &命令启动Redis时，启动项中已经设置好了Redis各个参数的默认值，Redis会按照这些设置的参数进行启动。但这些参数是可以在配置文件中进行修改的，修改后，需要在启动命令中指定要加载的配置文件，这样，配置文件中的参数值将覆盖原默认值。

```
[root@redis0S redis]# redis-server redis.conf
[root@redis0S redis]#
```

Redis已经给我们提供好了配置文件模板，是Redis安装目录的根目录下的redis.conf文件。由于刚刚对redis.conf配置文件做了修改，所以在开启Redis时需要显示指出要加载的配置文件。配置文件应紧跟在redis-server的后面。

2 连接前的配置

Redis是一个内存数据库服务器，就像MySQL一样，对其操作也需要通过客户端进行。若要使远程主机上的客户端能够连接并访问到服务端的Redis，则服务端首先要做如下配置。

绑定客户端IP

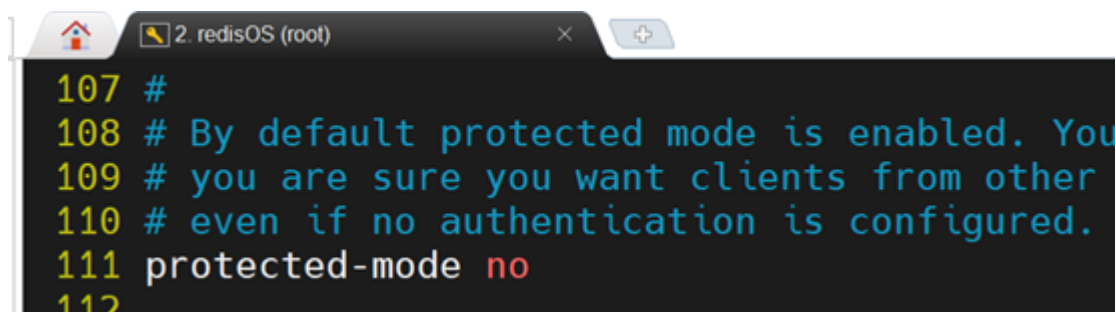
Redis可以通过修改配置文件来限定可以访问自己的客户端IP。

```
80 #
81 # IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO THE
82 # COMMENT OUT THE FOLLOWING LINE.
83 #
84 # You will also need to set a password unless you
85 # mode.
86 # ~~~~~
87 # bind 127.0.0.1 -:::1
88
```

以上设置后，只允许本机访问当前的Redis，其它主机均不可访问。所以，如果不想限定访问的客户端，只需要将该行注释掉即可。

关闭保护模式

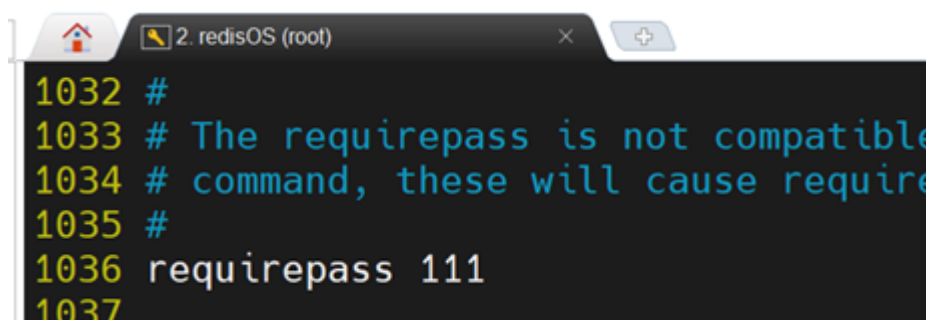
默认保护模式是开启的。其只允许本机客户端访问，即只允许自己访问自己。但生产中应该关闭，以确保其它客户端可以连接Redis。



```
107 #
108 # By default protected mode is enabled. You
109 # you are sure you want clients from other
110 # even if no authentication is configured.
111 protected-mode no
112
```

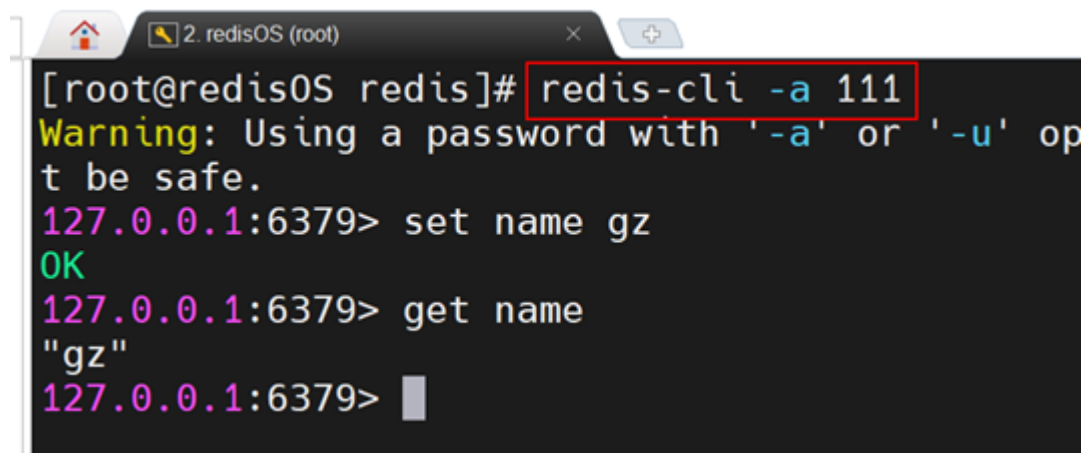
设置访问密码

为Redis设置访问密码，可以对要读/写Redis的用户进行身份验证。没有密码的用户可以登录Redis，但无法访问。访问密码的设置位置在redis.conf配置文件中，没有通过密码登录的用户，无法读/写Redis。默认是被注释掉的，没有密码。



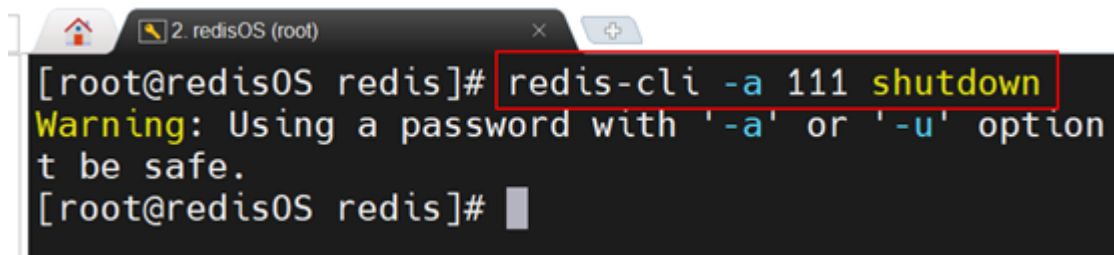
```
1032 #
1033 # The requirepass is not compatible
1034 # command, these will cause require
1035 #
1036 requirepass 111
1037
```

登录时使用密码



```
[root@redis0S redis]# redis-cli -a 111
Warning: Using a password with '-a' or '-u' op
t be safe.
127.0.0.1:6379> set name gz
OK
127.0.0.1:6379> get name
"gz"
127.0.0.1:6379>
```

退出时使用密码



```
[root@redis0S redis]# redis-cli -a 111 shutdown
Warning: Using a password with '-a' or '-u' option t be safe.
[root@redis0S redis]#
```

3 Redis客户端分类

Redis客户端也像MySQL客户端一样有多种类型：命令行客户端、图形界面客户端、Java代码客户端。

命令行客户端

Redis提供了基本的命令行客户端。打开命令行客户端的命令为redis-cli。



```
[root@redis0S redis]# redis-cli -h 127.0.0.1 -p 6379
127.0.0.1:6379>
```

-h用于指定要连接的Redis服务器的IP。-p用于指定要连接的Redis的端口号。若连接的是本机Redis，且端口号没有改变，保持默认的6379，则-h与-p选项可以省略不写。

图形界面客户端

Redis的图形界面客户端很多，其中较出名的是Redis Desktop Manager的客户端。不过，该软件原来是免费软件，从0.8.8版本后变为了商业化收费软件。

从官网（<https://resp.app/>）下载0.8.8版本使用。

RedisPlus下载地址：<https://gitee.com/MaxBill/RedisPlus>

Java代码客户端

所谓Java代码客户端就是一套操作Redis的API，其作用就像JDBC一样，所以Java代码客户端其实就是一个或多个Jar包，提供了对Redis的操作接口。

对Redis操作的API很多，例如jdbc-redis、jredis等，但最常用也是最有名的是Jedis。Jedis的Jar包可以通过在Maven工程中添加Jedis依赖获取。